ASCAL: AUTONOMOUS ATTITUDE SENSOR CALIBRATION

Chariya Peterson and John Rowe
Computer Sciences Corp. 10110 Aerospace Rd. Lanham-Seabrook, MD 20706

Karl Mueller and Nigel Ziyad
NASA Goddard Space Flight Center, Greenbelt, MD 20771

Abstract

In this paper, an approach to increase the degree of autonomy of flight software is proposed. We describe an enhancement of the Attitude Determination and Control System by augmenting it with self-calibration capability. Conventional attitude estimation and control algorithms are combined with higher level decision making and machine learning algorithms in order to deal with the uncertainty and complexity of the problem.

1 Introduction

The goal of our project is to enhance the degree of autonomy of the Attitude Determination and Control System (ADCS), enabling it to perform accurately without human intervention for an extended period of time. The approach is to evolve ADCS one step at a time into an autonomous system in a natural way dictated by actual needs. The purpose of this paper is to describe the first step in our program: the development of the Autonomous Attitude Sensor Calibration (ASCAL). The intention is to demonstrate ideas and concepts of on-board autonomy evolving from the existing control system, and not to develop another technique of attitude sensor calibration. A conventional ADCS uses data from available attitude sensors to estimate the attitude of the spacecraft. To meet mission pointing accuracy requirements, the attitude sensors must be calibrated for instrument biases, scale factors and misalignments immediately after launch and as needed thereafter. Traditionally, the calibration process is performed by attitude support specialists, often requiring elaborate procedures involving attitude consistency checks, data sampling and trending, and diagnosis expertise. A system that is able to perform all of these functions autonomously will have to deal with a large degree of uncertainty due to errors in the model parameters, incomplete model, measurement errors and human decision making. One of the new interdisciplinary areas currently emerging to tackle problems of this nature is the Intelligent Control Theory (Refs. 1-3) which combines conventional control theory with decision making and learning tools developed in the field of Artificial Intelligence. Following Tsypkin (Ref. 4), the necessity for applying learning arises in situations where a system must operate in conditions of uncertainty. Another active area of research is Hybrid Control Theory (Refs. 5, 6), which deals with systems that involve both continuous and discrete data structures. The discrete data may arise from sudden changes in the physical systems, from singularities in an incomplete dynamic model, from computer round off errors, or from actions controlled by higher level decision making. The discrete data often force the control system to make a choice and switch from one control law to another.

The system architecture adopted in this program has three layers: Execution, coordination, and planning. Each layer is organized further into a hierarchy of components, with the lowest level being the most precise and higher levels operating with less precise information and hence requiring an adaptive approach or learning approach. The choice between the adaptive or learning approach depends on the level of uncertainty of the problem. The adaptive approach may be sufficient for problems with less uncertainty. However, there are many different types of uncertainty. If an inconsistency exists in attitude estimation then it is unclear which onboard sensors are more reliable than the others. This is a more complex type of uncertainty. This type of uncertainty may have to be learned slowly through experience and is precisely the type of situation in which the learning process can prove most effective. Learning in this case takes place over a long time scale relative to the normal operation of the system.

The execution level is the lowest level. It involves conventional control algorithms and interfaces to the spacecraft via sensors and actuators. The highest level consists of planners and schedulers. In a mature system with more than one autonomous subsystem performing different functions, there may be only one planner and scheduler that manages tasks for all subsystems. The coordination level is the middle level, interfacing between the other two levels. This level consists of decision making tools, learning algorithms. Some of these tools may be used to substitute for conventional algorithms that are too costly or too sensitive to change or uncertainty. For instance, in this paper, we apply machine learning algorithms to control the calibration process instead of using batch or sequential processes to compute sensor residuals. The learning algorithm should be independent of the physical system and of any lower level process involved. More precisely, there are many state estimator algorithms to choose from. For each calibration task scheduled, only a few of these algorithms will be chosen. These choices should have no effect on the performance of the learning algorithm.

The layer and hierarchical structure of the architecture allows us to build on an existing control system, such as ADCS, step by step beginning with ASCAL which provides attitude sensor self-calibration functionality. As development progresses, higher level adaptation is made each time a new subsystem with new functionality is added to ADCS, such as gyroscope self-calibration functionality. The new subsystem can be operated and tested independent of previously developed subsystems.

Sensor calibration problems can be viewed as a dynamical system with uncertainty in the measurement model parameters. There are several algorithms for sensor calibration (Refs. 7-10). The choice of algorithm depends on the type of sensors being considered. Typically, it is left to the attitude experts to select appropriate methods for the task. However, to demonstrate the ideas and concepts of ADCS enhanced autonomy, we will focus on only one algorithm. In a later stage of development, when the concept of self-calibration has matured, additional algorithms may be added as new subsystems in the hierarchy. Expert knowledge on algorithm selection would be coded as rules in a rule-based system in the mid-level. The rule-base will select an appropriate algorithm when a calibration task is scheduled.

An automated system such as ASCAL is useful for mission cost reduction. It automatically performs routine monitoring and trending and stores experts' knowledge of sensor and instrument calibration to be reused for future events. Moreover, ASCAL may be useful for constellation of satellites, each having similar pointing requirements. Our future extension is to apply the same architecture described here to other flight software such as orbit determination and navigation systems, tracking, and formation flying.

This paper is organized as follows. The main architecture of the system is described in Section 2. The main focus of this paper, the calibration component, is described in Section 3. The technology used in the calibration component is a heuristic learning automaton. The prioritization for the calibration process is based on the Local Dempster-Shafer theory developed in [1]. This is described in Section 4. The Coordinator and Planner level are discussed in Section 5 and 6 respectively.

2 ASCAL Architecture

Figure 1 shows the architecture of ASCAL. The execution level consists of an attitude estimator and predictor. The coordination level determines which sensor parameters need adjustment, what should their upper and lower bounds be, and which algorithms are appropriate. This level also includes the learning component in the calibration process. The planning level plans and schedules calibration tasks, making sure that computing resources are available and avoiding possible conflicts with other tasks.

It is natural to consider extended state vectors consisting of an attitude vector and erroneous sensor parameters. However, this will generally introduce additional non-linearity into the models and could make the problem intractable or too costly to run on-board. To minimize the computational cost, we apply machine learning techniques to adjust these parameters guided by past experience. Attitudes and errors are computed each time sensor parameters are adjusted. Each cycle of the computation contributes new information on the convergence of the solution. This knowledge will affect the way these parameters are adjusted.

Naively, attitude accuracy is monitored by estimating attitude using different combinations of gyros and attitude sensors, uncalibrated versus calibrated. The attitude residuals obtained from the computed attitudes are predicted using a conventional prediction algorithm. When it is discovered that the attitude residual will exceed a threshold sometime in the future, it means there is an inconsistency in the estimated attitudes. The attitude inconsistencies are then diagnosed and one or more calibration goals are created. These goals are expressed as which measurement parameters need adjustment, the range of adjustment and the most appropriate calibration algorithm. The calibration process is then planned and scheduled. In a spacecraft where one or more sensors need regular calibration, or where computing resources are limited, the predictor may be replaced by a periodic schedule managed by the planner/scheduler component. The calibration process is iterative, where the erroneous measurement parameters are adapted on the basis of system experience in such a way that the attitude inconsistencies converge to zero.

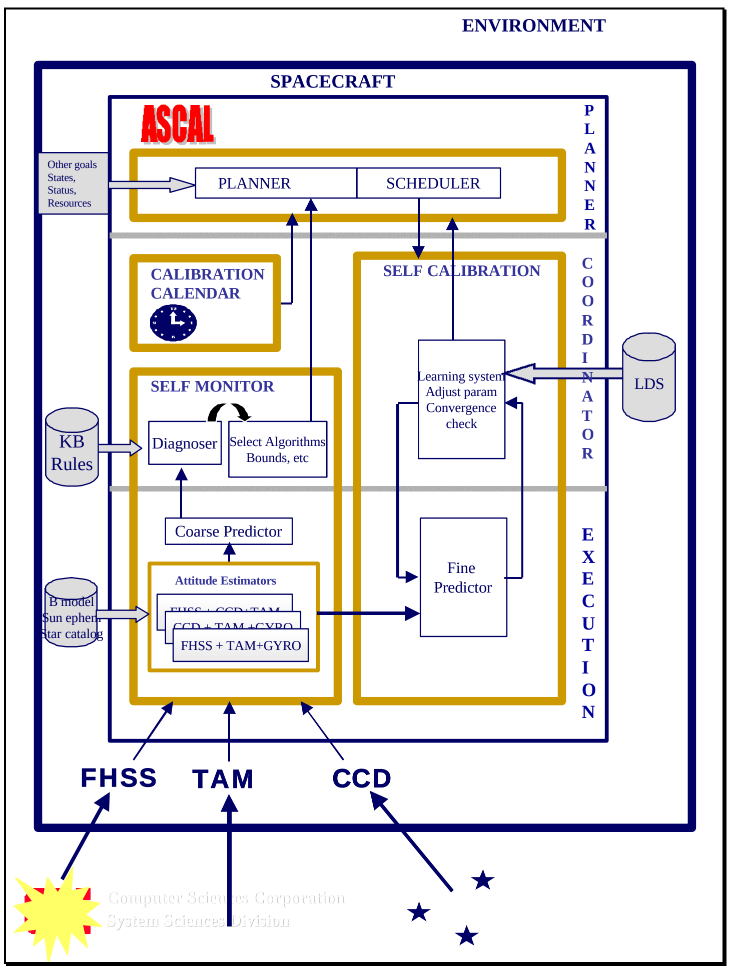

Figure 1 ASCAL architecture

3. Estimator and Predictor

When a calibration process is scheduled, the coordinator will set a goal following a guideline stored in its knowledgebase, perhaps as a set of rules. A typical goal would be to calibrate a certain set of parameters. The calibration procedure depends on the type of sensors on-board. If there are enough redundant sensors the standard technique is to compute attitudes from a few different sets of sensors and compare the results. If the pairwise difference between these attitudes have zero mean, then there is no inconsistency, and all of the sensors are accurate (relative to each other). Generally, there are one or more sensors that are used as standard. They are the ones that have already been calibrated, or the ones with higher accuracy. We will call a set of sensors used in an attitude estimation process a *test set*. Generally, one or more of the test sets contain sensors to be calibrated, and at least one of the test set contains accurate sensors. If there are no redundant sensors, or not enough available sensors to create at least two test sets, then the calibration procedure usually involves more in depth analysis. In this paper, we assume there is at least one sensor with high accuracy, such as a Charge Coupled Device (CCD) star tracker, enabling us to calibrate other sensors against them. Such a sensor is frequently chosen as the standard frame of reference and generally does not need calibration. In this paper, we assume that there is such sensor on-board.

Before the calibration process starts, a number of test sets are identified, with at least one of the test sets containing the sensor(s) to be calibrated and the other test sets containing the standard sensor, calibrated gyros, or other high accuracy sensors. The coordinator, via its rule-base component, will also select a suitable estimator algorithm, for instance an attitude dynamic model and a measurement model for each selected sensor.

Let a denote a test set, x_a the attitude vector computed using measurements from all sensors in a. The attitude dynamics and the corresponding measurement model are

$$\begin{aligned} \dot{x}_a &= f(x_a(t)) + u_a(t) \\ z_{a_i,k} &= G_{a_i}(p_{a_i}, x_a(t_k)) + w_{a_i}(t_k), \end{aligned} \tag{1}$$

where a_i is a sensor in a, p_{a_i} is its model parameter vector. Note that, in this algorithm, each p_{a_i} is assumed constant during each estimation cycle. They are not members of state variables, however, their values will be adjusted by the learning system described in the next section.

In the following, we give a simple example of a state estimator and trend predictor to demonstrate how the learning system can be used in a calibration process. The inconsistency trend between attitude vectors associated with two different test sets *a* and *b* is the difference $T_{ab} = x_a - x_b$. The state space model for the inconsistency trend and its slope S_{ab} are

$$\begin{aligned} T_{ab}(t_{k+1}) &= T_{ab}(t_k) + S_{ab}(t_k) + n_{ab}(t_k) \\ S_{ab}(t_{k+1}) &= S_{ab}(t_k) + w_{ab}(t_k) \end{aligned}$$

Define a new state vector

$$X_{ab} = [T_{ab} - w_{ab} \quad S_{ab}]'$$

Then we have the following state-space model

$$\begin{aligned} X_{ab}(t_{k+1}) &= AX_{ab}(t_k) + u_{ab}(t_k) \\ T_{ab}(k) &= HX_{ab}(k) + w_{ab}(k+1) \end{aligned} \tag{2}$$

with

$$n_{ab}(t_k) = w(t_{k+1}) - w(t_k), \; u_{ab} = [w_{ab} \quad w_{ab}]',$$

and

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \; H = [1 \quad 0]$$

In system (2) T_{ab} plays the role of output vector with preferred state $T_{ab,0}$ = 0. All of the process and measurement noises are assumed to be white Gaussian with zero mean. The systems (1) and (2) become a two-stage problem. Given a set of sensor parameters, attitudes are computed either by a batch least square or a sequential method. System (2) predicts the inconsistency trend. Here we write it as a single step predictor, but a multiple step predictor can also be done.

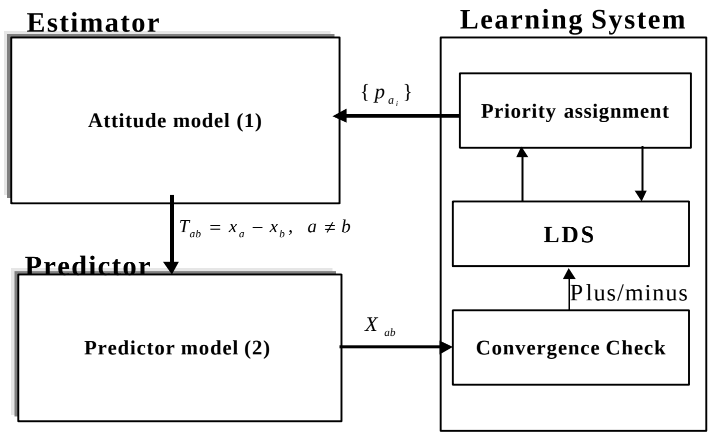

Figure 2. ASCAL Learning System

4 Learning Systems

The heart of a learning system is the learning algorithm which is the mechanism used to adapt the probability distribution. Based on the environment response and the action selected by the system at time t, it generates $p(t+1)$ from $p(t)$. There are two levels of learning in ASCAL. When a calibration goal is set, the coordinator must determine the sensors, algorithms, and parameter ranges needed to initialize the calibration process. This selection is based on the past experiences. In particular, the parameter ranges are chosen in such a way that the region is void of any singularity and at least one solution exists. This knowledge can be given *a priori* by attitude experts, and maintained by a learning algorithm. The second level of learning is in the calibration process, where attitude residuals are computed, convergence tested, and parameters adjusted sequentially. We assume that an appropriate metric is defined on the state space. The selection of parameter adjustment is a learning process based on the rate of convergence (or divergence) of the attitude residuals during the previous two (or more) cycles. Assume there are n sensor parameters to be adjusted, and each parameter can be increased or decreased by a fixed quantity. This corresponds to $H = \sum_{i=0}^{n} 2^i \binom{n}{i}$ possible actions, where each action is a set of parameters, each associated with a + or – sign to denote if it is increased or decreased. For instance, an action corresponding to an increase in a and decrease in b is represented by the signed set $\{a_+, b_-\}$. These H actions are prioritized by a probability or belief vector given by the Local Dempster-Shafer (LDS) (Ref s 11, 12).

To get a feel for the learning algorithm based on LDS, we will now describe a simpler algorithm based on the Dempster-Shafer (DS) theory (Refs. 13, 14), modified to suit our calibration problem. For a more in depth discussion of the LDS theory see Ref. 12. DS theory is defined on a set of n elements. A mass function on the action set H is a probability function that assigns a degree of belief to each action. More precisely, the mass function satisfies the following conditions

$$\sum_{A \supseteq H} m(A) = 1, \text{ for } A \neq \emptyset \quad \text{and} \quad m(\emptyset) = 0$$

Two mass functions m_1 and m_2 on H can be combined into a single mass function $m_1 \otimes m_2$ by the Dempster composition rule:

$$m_1 \otimes m_2(A) = \sum_{B \cup C = A} m_1(B) m_2(C) \Big/ \Big(1 - \sum_{B \cup C = \emptyset} m_1(B) m_2(C)\Big), \quad \text{for } A \neq \emptyset$$

$$m_1 \otimes m_2(\emptyset) = 0.$$

The belief function associated to the mass function *m* is defined to be the cumulative probability distribution on *H*:

$$b : H \rightarrow [0,1];\; b(A) = \sum_{B \supseteq A} m(B)$$

where the union between two signed sets is defined as the union of all signed elements, followed by removing every subset of the form $\{a_+, a_-\}$ for some parameter a. The belief function is used to prioritize the actions for the learning algorithm. If an action is chosen and the resulting attitude residuals decrease with a faster rate or increase with a slower rate, then the system reprioritizes by applying the positive learning algorithm described in Ref 12. This will strengthen the previous prioritization. Conversely, if the previously chosen action is performed in the opposite manner, then the system reprioritizes by applying negative learning algorithm, which will lessen the degree of belief on the failed action.

In general, a learning system may have a hierarchical structure. In this case, the selection of the action set should also have a hierarchical structure. To support this structure, a hierarchical flavor of DS theory can be defined in a natural way. The action selection is performed in a sequence of steps. First, a highest level in the hierarchy of the action set is selected, followed by a lower action. This procedure is followed until the last level of the action set *H*. This hierarchical structure will clearly reduce the size of the search space, and hence enhance the performance of the automata.

The learning process discussed above is the simplest application of the (modified) DS theory to learning automata. In practice this algorithm can be enhanced in several different ways to increase the performance and robustness of the learning system. Our possible future research topics in this areas are: Localization of the action space (*H*) by applying LDS theory instead of DS theory. This will reduce the complexity of the search and increase the performance of ASCAL. Instead of keeping the step size of parameter modification constant, we may consider it as a function of the rate of convergence computed from the previous cycles. The function that works will guarantee the convergence of the solution. The use of hierarchical or multilevel learning systems accelerates the learning process (more so for the initial rate of learning) and simplifies the structure of the learning system. The learning system discussed above is an active research area with many applications in intelligent and hybrid control problems.

5 Coordinator

In some sense, the coordinator is a process manager whose responsibility is to monitor the physical subsystem it is responsible for, i.e. the ADCS, and predict if any problem, i.e. an attitude inconsistency, will occur. If a problem is predicted, the coordinator will identify the source of the problem and create goals to solve it.

The responsibility of the coordinator consists of two parts: monitoring/diagnosis and pre-calibration. The monitoring/diagnosis components monitors the state of health of the ADCS by periodically computing and trending relative attitude residuals using multiple test sets. When an attitude inconsistency is predicted, the diagnosis component determines which sensor parameters are likely to be unreliable based on attitude data that displays the trends. The result of the diagnosis is the degree of unreliability, a probabilistic quantity, assigned to each sensor parameter involved in the trending process. Underlying the diagnosis process is the uncertainty handler based on the LDS theory, (Ref 11). When this is done, the coordinator creates goal to calibrate the problematic parameters, and submits the goal to the planner.

When a calibration process is scheduled the pre-calibration tasks begin. First, based on the degree of unreliability, a collection of test sets is formed, and the bounds for the sensor parameters are computed. Based on the sensors involved, attitude dynamics and measurement models are selected, and a state-space system is defined for each test set. Finally, the coordinator also determines any *a priori* knowledge the calibration process may need, including the initial probability distribution for the learning system to use as priority assignment in the learning process. It is convenient to use the degree of unreliability as the initial probability distribution. However, other expert knowledge the system may have can be combined with the degree of unreliability using a modified Dempster combination rule. The coordinator performs these tasks using decision making capability such as a rule-base.

6 Planner & Scheduler

This component may be responsible for several autonomous systems. For ASCAL, the planner/scheduler is responsible for scheduling sensor calibration. It should be aware of available sensors, i.e. those with target in field of view, and related resources. This means the spacecraft must be sufficiently equipped with a star catalog, and Sun, Earth, and Moon ephemerides. The system must also be able to perform some maneuver planning needed for gyroscope calibration, or to sample selected targets throughout the field of view.

In this version of ASCAL, the calibration process uses live data from attitude sensors on-board, to avoid dealing with attitude history data management which is a formidable problem of its own. However, as a trade off, the planner will have to be smart enough to avoid conflicts among spacecraft activities, to manage resources such as available sensors and computer time. A simple solution is to find a quiet window of time when there is no important activity on-board and devote all attention to the calibration process.

7 Conclusion and Implementation Status

This study is the first phase of our program to extend the degree of autonomy of on-board flight software. The consequences of failure are catastrophic for an attitude control system. If the attitude control system fails for even a brief period, the spacecraft may tumble, pointing the solar arrays away from the sun, antennas away from the earth, and sensitive instrumentation in a potentially damaging direction. Such a control system failure may or may not be recoverable. Nonetheless, virtually all spacecraft have fully autonomous, onboard attitude control. Failure to properly calibrate the sensor parameters would lead to inaccuracies in attitude estimation, and would in turn lead to attitude control system failure. Sensor calibration is traditionally done from the ground, because the standard procedures and algorithms are storage and computationally intensive. In this paper, we propose a non traditional approach, using learning automata and heuristic priority assignment to adjust sensor parameters until all inconsistencies converge to within an acceptable limit. Human intervention is called for if this process does not converge, and if the diagnoser cannot resolve the problem. In this case, the lessons learned should be added into the knowledgebase for future use. It is important to design the learning algorithms so that they are independent of the sensors being calibrated or of changes in the environment. This is key for autonomous attitude sensor calibration in future missions.

The next natural step towards higher level on-board automation is to add data management capability to ASCAL. Calibration process can be performed using historical data without disturbing other activities, except computer resources. To archive measurement data for the calibration process ahead of time would require, the coordinator can be augmented with a data processing component. It is responsible for data pre or post processing, data smoothing, and/or shifting. Generally, measurement data are sensitive to some spacecraft's activities such as maneuvering. The planner/scheduler must be aware of these activities. With this knowledge obtained from the planner/scheduler, the data processor may avoid the disturbed data, so that relatively clean data for the past, say 24 hours, may be stored and ready to be used when a calibration process is scheduled. This problem suggests that high level autonomy is necessary for autonomy system development such as ASCAL.

Other possible future development is autonomous orbit determination and control, orbit keeping, maneuvering, and formation flying. Machine learning approach described in this paper is a generic tool that is likely to be useful in these applications.

References

1. P. J. Antsaklis, and K. M. Passino, eds., *An Introduction to Intelligence and Autonomous Control*, Norwell, MA: Kluver Academic Publishers, 1993.

2. P. J. Antsaklis, K. M. Passino, and S. J. Wang, *An Introduction to Autonomous Control Systems*, Special Issue on Intelligence Control, IEEE Control Systems Magazine, 11, 4, 5-13 1991.
3. P. J. Antsaklis, K. M. Passino, and S. J. Wang, *Towards Intelligent Autonomous Control System: Architecture and Fundamental Issues*, Journal of Intelligent and Robotic Systems, 1, 4, 315-342, 1989.
4. Y. T. Tsypkin, *Foundation of the Theory of Learning Syste*ms, Academic Press 1973.
5. P. J. Antsaklis, M. D> Lemmon, and J. A. Stiver, *Hybrid System Modeling and Event Identification*, Proc of the 29th Allerton Conf. On Communication, Control, and Computing. 64-73, University of Illinois at Urbana-Champaign, Sep. 1993.
6. A. Gollu, and P. Varaiya, *Hybrid Dynamical Systems*, Proc. Of the 28th Conf on Decision and Control, 2708-2712, Tampa, FL, Dec. 1989.
7. M. D. Shuster, D. M. Chitre, and D. P. Niebur, *Inflight Estimation of Spacecraft Attitude Sensor Accuracies and Alignments*, J. of Guidance and Control, 5, 4, 1982.
8. M. D. Schuster, *In-Flight Estimation of Spacecraft Sensor Alignment*, Advances in the Astronautical Sciences, 72, 1990.
9. G. J. Bierman and M. D. Schuster, *Spacecraft Alignment Estimation*, 27th IEEE Conference on Decision and Control, Austin, TX, Dec. 7-9 1988.
10. J. E. Keat, *Gyro Calibration Analysis for the High Energy Astronomy Observatory-A (HEAO-A)*, Computer Sciences Corporation, CSC/TM-77/6082, June 1977.
11. C. Sary, C. Peterson, J. Rowe, K. Mueller W. Truszkowski and N. Ziyad, *Trend Analysis for Spacecraft Systems Using Multimodal Reasoning*, 152-158, AAAI Spring Symposium, Stanford, Palo Alto, CA, Mar. 1998.
12. C. Peterson, K. Mueller, *Local Dempster Shafer Theory*, submitted to the J. of Experimental and Theoretical Artificial Intell
13. G. A. Shafer, *Mathematical Theory of Evidence*, Princeton U. Press, 1976.
14. A. P. Dempster, *Upper and Lower Probabilities Induced by Multivalued Mappings*, Annals of Math. Stat. 38, pp. 325-329, 1967.